

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2000-235542

(43)Date of publication of application : 29.08.2000

(51)Int.Cl.

G06F 13/14

G06F 12/02

G06F 12/10

G06F 13/28

G06F 13/36

G11B 20/10

(21)Application number : 11-037631

(71)Applicant : SONY CORP

(22)Date of filing : 16.02.1999

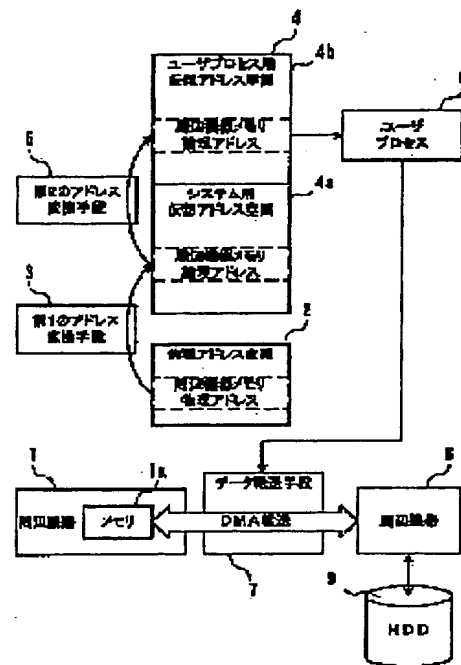
(72)Inventor : KON MASASHI
NAKAMURA RYUICHI

(54) DATA PROCESSOR AND RECORDING MEDIUM

(57)Abstract:

PROBLEM TO BE SOLVED: To speedily transfer data transfer between plural peripheral equipments while utilizing the functions of a file management system.

SOLUTION: A 1st address conversion means 3 converts a physical address of a peripheral equipment 1 into a logical address of a system virtual address space 4a formed in a virtual address space 4. A 2nd address conversion means 5 converts the logical address converted by the means 3 into a logical address in a user processor virtual address space 4b corresponding to a specific user process 6 executed on an OS. A data transfer means 7 executes data transfer based on a direct memory access between a memory 1a of the equipment 1 and another peripheral equipment 8 based on a data transfer request outputted from the user process 6 and specifying the logical address converted by the means 5.



LEGAL STATUS

[Date of request for examination]
[Date of sending the examiner's decision of rejection]
[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]
[Date of final disposal for application]
[Patent number]
[Date of registration]
[Number of appeal against examiner's decision of rejection]
[Date of requesting appeal against examiner's decision of rejection]
[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

50490430 u500

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2000-235542

(P2000-235542A)

(43) 公開日 平成12年8月29日 (2000.8.29)

(51) Int.Cl. ⁷	識別記号	F I	テーマコード*(参考)
G 0 6 F 13/14	3 2 0	G 0 6 F 13/14	3 2 0 H 5 B 0 0 5
12/02	5 7 0	12/02	5 7 0 M 5 B 0 1 4
12/10		12/10	H 5 B 0 6 0
13/28	3 1 0	13/28	3 1 0 M 5 B 0 6 1
13/36	3 1 0	13/36	3 1 0 B 5 D 0 4 4

審査請求 未請求 請求項の数 3 O L (全 11 頁) 最終頁に続く

(21) 出願番号 特願平11-37631

(22) 出願日 平成11年2月16日 (1999.2.16)

(71) 出願人 000002185

ソニー株式会社

東京都品川区北品川6丁目7番35号

(72) 発明者 昆 雅士

東京都品川区北品川6丁目7番35号 ソニ

ー株式会社内

(72) 発明者 中村 隆一

東京都品川区北品川6丁目7番35号 ソニ

ー株式会社内

(74) 代理人 100092152

弁理士 服部 毅巖

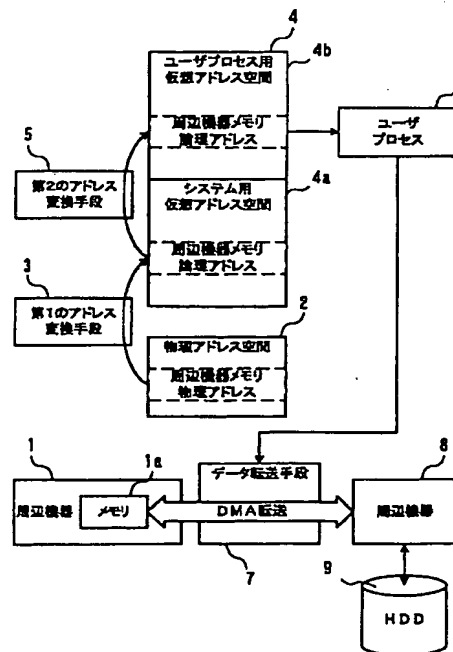
最終頁に続く

(54) 【発明の名称】 データ処理装置及び記録媒体

(57) 【要約】

【課題】 ファイル管理システムの機能提供を受けつ、周辺機器間的高速なデータ転送を可能とする。

【解決手段】 第1のアドレス変換手段3は、周辺機器1のメモリ1aの物理アドレスを、仮想アドレス空間4に設けられたシステム用仮想アドレス空間4aの論理アドレスへ変換する。第2のアドレス変換手段5は、第1のアドレス変換手段3による変換後の論理アドレスを、OS上で実行されている特定のユーザプロセス6に対応するユーザプロセス用仮想アドレス空間4b内の論理アドレスへ変換する。データ転送手段7は、ユーザプロセス6からの第2のアドレス変換手段5によって変換された論理アドレスを指定したデータ転送要求に基づいて、周辺機器1のメモリ1aと他の周辺機器8との間のダイレクトメモリアクセスによるデータ転送を行う。



【特許請求の範囲】

【請求項 1】 メモリを搭載した周辺機器が接続されたデータ処理装置において、

前記周辺機器のメモリの物理アドレスを、システム制御用の仮想アドレス空間内の論理アドレスへ変換する第 1 のアドレス変換手段と、

前記第 1 のアドレス変換手段による変換後の論理アドレスを、オペレーティングシステム上で実行されている特定のユーザプロセスに対応する固有の仮想アドレス空間内の論理アドレスへ変換する第 2 のアドレス変換手段と、

前記ユーザプロセスからの前記第 2 のアドレス変換手段によって変換された論理アドレスを指定したデータ転送要求に基づいて、前記周辺機器のメモリと他の周辺機器との間のダイレクトメモリアクセスによるデータ転送を行うデータ転送手段と、

を有することを特徴とするデータ処理装置。

【請求項 2】 前記周辺機器は、バスブリッジチップを介してプロセッサ側と接続された二次バス上に実装されており、

前記データ転送手段は、前記二次バス上に実装された他の周辺機器に対してデータ転送を行うことを特徴とする請求項 1 記載のデータ処理装置。

【請求項 3】 メモリを搭載した周辺機器を対象としたデータ転送を行うためのデータ転送制御プログラムを記録した記録媒体において、

周辺機器のメモリの物理アドレスを、システム制御用の仮想アドレス空間内の論理アドレスへ変換する第 1 のアドレス変換手段、

前記第 1 のアドレス変換手段による変換後の論理アドレスを、オペレーティングシステム上で実行されている特定のユーザプロセスに対応する固有の仮想アドレス空間内の論理アドレスへ変換する第 2 のアドレス変換手段、

前記ユーザプロセスからの前記第 2 のアドレス変換手段によって変換された論理アドレスを指定したデータ転送要求に基づいて、前記周辺機器のメモリと他の周辺機器との間のダイレクトメモリアクセスによるデータ転送を行うデータ転送手段、

としてコンピュータを機能させるためのデータ転送制御プログラムを記録したコンピュータ読み取り可能な記録媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は複数のボード間のデータ転送を行うデータ処理装置に関し、特に大量のデータ転送が必要となるデータ処理装置に関する。

【0002】

【従来の技術】コンピュータの性能向上により、従来は専用の装置で行っていた作業を汎用的なコンピュータで置き換えることが可能となっている。そのような機器の

1 つにビデオ編集機がある。すなわち、VTR (Video Tape Recorder) などの画像や音声の編集作業を、コンピュータを用いて行うことが可能となっている。その場合、画像情報や音声情報をコンピュータのハードディスク装置に格納し、コンピュータのアプリケーションプログラムを用いて編集作業を行う。このようなビデオ編集システムを、ノンリニア編集機という。

【0003】ノンリニア編集機は、画像の変形・合成が自由にできることや、編集を繰り返しても画質の劣化がないという点において、従来の編集システムより優れている。例えば、実写の画像に CG (Computer Graphics) を合成するのも容易である。ところで、ノンリニア編集機では、動画データなどをハードディスク装置に格納しているため、ハードディスクから汎用バス（例えば PCI (Peripheral Component Interconnect) バス）を介して、他のボード（例えば、CODEC (coder-decoder) ボード）へ転送する必要が生じる。このようにボード間でデータ転送を行う場合、基本的には、ノンリニア編集機中のパーソナルコンピュータ部分（ノンリアルタイム制御部）内のシステムメモリを介在させて、データ転送が行われる。すなわち、データソースとなるボードからシステムメモリへの転送、システムメモリからデータターゲットとなるボードへの転送の 2 回の転送処理を実行する。この方法を第 1 の従来方法とする。

【0004】また、OS (Operating System) などのソフトウェアの影響を受けないで、直接ボード間でデータを転送する方法もある。これは、DMA (Direct Memory Access) 転送を用いたものである。DMA 転送は、基本的に主記憶装置（システムメモリ）から二次記憶装置にデータ転送を行う際に用いられる機能であるが、これをボード間のデータ転送に応用することで、データ転送の高速化が可能となる。この方法を第 2 の従来方法とする。

【0005】

【発明が解決しようとする課題】しかし、従来の方ではいずれの方法においても以下のような問題点があった。

【0006】第 1 の従来方法では、ソフトウェア（特に OS）が提供するサービス（特にファイル管理やメモリ管理など）を受けられる反面、2 回データ転送を行う必要があるため、性能が低くなってしまふ。

【0007】第 2 の従来方法では、直接伝送するため転送性能は高いが、ソフトウェア（特に OS）が提供するサービス（特にファイル管理）を受けられない。従って、ファイル管理などの必要のないデータ転送（例えば、画像データをグラフィック制御ボードへ転送する場合）では有効であるが、ハードディスクへ保存する場合には適用できない。

【0008】そのため、ソフトウェア（特に OS）のサービスが必須となるシステムにおいて第 1 の従来方法を利用することとなる。ところが、ノンリニア編集機など

では映像データを転送する関係上非常に高い転送性能が要求されるため、第1の従来方法では不適切である。すなわち、ノンリニア編集機のような映像処理を行うシステムにおいては、ソフトウェア（特にOS）のサービスを利用でき、かつボード間的高速転送が可能であることが望まれている。

【0009】本発明はこのような点に鑑みてなされたものであり、ファイル管理システムの機能提供を受けつつ、周辺機器間的高速なデータ転送が可能なデータ処理装置を提供することを目的とする。

【0010】

【課題を解決するための手段】本発明では上記課題を解決するために、メモリを搭載した周辺機器が接続されたデータ処理装置において、前記周辺機器のメモリの物理アドレスを、システム制御用の仮想アドレス空間内の論理アドレスへ変換する第1のアドレス変換手段と、前記第1のアドレス変換手段による変換後の論理アドレスを、オペレーティングシステム上で実行されている特定のユーザプロセスに対応する固有の仮想アドレス空間内の論理アドレスへ変換する第2のアドレス変換手段と、前記ユーザプロセスからの前記第2のアドレス変換手段によって変換された論理アドレスを指定したデータ転送要求に基づいて、前記周辺機器のメモリと他の周辺機器との間のダイレクトメモリアccessによるデータ転送を行うデータ転送手段と、を有することを特徴とするデータ処理装置が提供される。

【0011】このようなデータ処理装置によれば、周辺機器のメモリの物理アドレスが、第1のアドレス変換手段によって、システム制御用の仮想アドレス空間内の論理アドレスへ変換される。次に、第1のアドレス変換手段による変換後の論理アドレスが、第2のアドレス変換手段によって、オペレーティングシステム上で実行されている特定のユーザプロセスに対応する固有の仮想アドレス空間内の論理アドレスへ変換される。そして、ユーザプロセスからの第2のアドレス変換手段によって変換された論理アドレスを指定したデータ転送要求が出されると、データ転送手段によって、周辺機器のメモリと他の周辺機器との間のダイレクトメモリアccessによるデータ転送が行われる。

【0012】

【発明の実施の形態】以下、本発明の実施の形態を図面を参照して説明する。図1は、本発明の原理構成図である。本発明のデータ処理装置には、メモリ1aを搭載した周辺機器1が接続されている。メモリ1aには、物理アドレス空間2内のアドレスが割り当てられている。第1のアドレス変換手段3は、周辺機器1のメモリ1aの物理アドレスを、仮想アドレス空間4に設けられたシステム用仮想アドレス空間4aの論理アドレスへ変換する。第2のアドレス変換手段5は、第1のアドレス変換手段3による変換後の論理アドレスを、OS上で実行さ

れている特定のユーザプロセス6に対応するユーザプロセス用仮想アドレス空間4b内の論理アドレスへ変換する。データ転送手段7は、ユーザプロセス6からの第2のアドレス変換手段5によって変換された論理アドレスを指定したデータ転送要求に基づいて、周辺機器1のメモリ1aと他の周辺機器8との間のダイレクトメモリアccessによるデータ転送を行う。図の例では、周辺機器8にハードディスク装置（HDD）9が接続されており、メモリ1aのデータをHDD9に書き込むような処理が行われる。

【0013】このようなデータ処理装置によれば、周辺機器1のメモリ1aのアドレスが、ユーザプロセス用仮想アドレス空間内に定義される。そのため、ユーザプロセス6から周辺機器1のメモリ1aのアドレスを直接指定したデータ転送要求が可能となる。ユーザプロセス6からのアドレス指定に基づくデータ転送であれば、ユーザプロセスが実行されているOS上の機能を楽しむことができる。例えば、図の例のようにHDD9にデータを格納する場合、転送されたデータはOSのファイル管理システムの管理下で保存される。また、データ転送手段7が周辺機器1のメモリ1aから直接（システムのメインメモリを介さず）DMA転送を行っているため、高いデータ転送効率を得ることができる。

【0014】本発明をノンリニア編集機に適用することで、画像データ的高速転送が可能となる。そこで、ノンリニア編集機に本発明を適用した場合の例を以下に示す。なお、以下の例では、OSとしてWindowsNT（マイクロソフト社の登録商標）を用いた場合の具体例を示しながら説明する。

【0015】図2は、ノンリニア編集システムのハードウェア構成例を示す図である。CPU（Central Processing Unit）11は、他の周辺機器やシステムメモリ13を制御することでOSの機能を実行するとともに、OSに実装されたデバイスドライバや、OS上で実行されるアプリケーションソフトウェアを実行する。なお、このCPU11は、アドレス変換テーブルを参照することで、仮想メモリ空間の論理アドレスをシステムメモリ13やCODECボード41内のメモリ41aの物理アドレスに変換する機能を有している。

【0016】PCIバスコントローラ12は、CPU11側からPCIバス21を制御するためのコントローラである。システムメモリ13は、システムの主記憶装置であり、CPU11で実行されるプログラムや、処理に必要な各種データが一時的に格納される。

【0017】PCIバス21は、PCIバスブリッジ31～33によって、複数のPCIバス22～24に接続されている。複数のPCIバス21～24は、CPU11と周辺機器との間のデータ転送、及び周辺機器間のデータ転送を行うためのバスである。

【0018】PCIバスブリッジ31～33は、個別の

PCIバスを互いに電氣的に接続する働きを有する。また、一方のPCIバス内の装置間でのデータ転送が行われた場合には、他方のPCIバスには信号を出力しないような制御も行う。

【0019】CODECボード41は、PCIバス22、ローカルCPUバス51及びDAV(Digital Audio Visual)バス52に接続されている。このCODECボード41は、画像データを圧縮・伸張する機能を有する。また、CODECボード41にはメモリ41aが設けられており、このメモリ41a内に、圧縮・伸張等の

処理後の画像データを格納する。CODECボード41の出力映像はDAVバス52を介してDME(Distributed Management Environment)/SW(Switcher)ボード45に渡さる。

【0020】SCSI HBA(Small Computer System Interface Host Bus Adapter)カード42は、PCIバス22に接続されており、PCIバス22を介して送られてきたデータをHDD43に格納したり、HDD43内のデータをPCIバス22を介して他の機器へ転送したりする。SCSI HBAカード42は、データ転送の際にPCIバスのホストとなり、DMA転送を制御することができる。

【0021】HDD43は、映像データなどのデータを保持する。ローカルCPUボード44は、PCIバス23とローカルCPUバス51とに接続されており、ローカルCPUを搭載している。ローカルCPUは、ローカルCPUバス51を介して接続された機器(図の例では、CODECボード41)の動作をリアルタイムに制御する。

【0022】DME/SWボード45は、PCIバス24に接続されているとともに、複数の入出力デバイスに接続されている。DME/SWボード45は、映像データに対して効果をかけたり出力先を切り替える処理を行う。また、このボードで設定した映像入力にCODECボード41にDAVバス52を介して入力される。また、CODECボード41から送られた画像データは、DME/SWボード45を中継して映像出力となる。

【0023】ローカルCPUバス51は、ローカルCPUが制御を行うボードすべてとローカルCPUボード44との間を接続している。そして、ローカルCPUが他の周辺機器を制御するためのコマンド送信等を使用される。

【0024】DAVバス52は、ディジタル信号化された映像データや音声データを転送するためのバスであり、映像や音声データを処理するボードすべてに接続されている。図の例では、CODECボード41とDME/SWボード45とがDAVバス52に接続されている。

【0025】なお、図に示した要素以外にもノンリニア編集機はオーディオボードなどを有しているが、この例

では映像に関する説明に絞ったため、図2には示していない。

【0026】次に、アップロード処理の概要について説明する。ここで、アップロードとは、入力デバイス(カメラやビデオデッキなど)から、映像や音声をHDD43に記録することである。アップロード時のデータの流れを簡単に説明すると、以下のようになる。

【0027】DAVバス52を介して入力された映像は、CODECボード41で圧縮(または非圧縮)され、CODECボード41内のメモリ41aに蓄積される。CODECボード41上のメモリ41a内のデータは、SCSI HBAカード42を介してHDD43に記録される。HDD43はOSによってフォーマットされており、記録されるデータは、OSのファイル管理システムに従った内容としてHDD43に記録される。

【0028】さらに、ダウンロード処理の概要について説明する。ここで、ダウンロードとは、HDD43に記録されている映像や音声データを、システムに接続された記録デバイス(ビデオデッキやモニタ)に再生することである。ダウンロード時のデータの流れを簡単に説明すると、以下のようになる。

【0029】HDD43に記録されているデータは、OSからはファイルとして管理・処理される。記録した映像データをCODECボード41を使い再生する場合、HDD43からファイルの内容を読み出す。読み出したデータは、CODECボード41上のメモリ41aに置く。そして、CODECボード41が圧縮データの場合、伸長処理を行いDAVバス52に流す。そのデータはDME/SWボード45で受け取られる。DME/SWボード45は、受け取った映像データを出力デバイスに流す。

【0030】次に、データのアップロードやダウンロードが行われる際に、CODECボード41とSCSI HBAカード42との間でのデータ転送を行うための処理について説明する。

【0031】ところで、従来の方法では、例えばアップロードの際には、CODECボード41のメモリ41aに保持されたデータをシステムメモリ13に転送し、その後、SCSI HBAカード42にそのデータを転送していた。すなわち、2回の転送によってCODECボード41のデータがHDD43に記録されていた。というのは、HDD43とのやり取りにおいて、直接CODECボード41上メモリ41aの物理アドレスを指定する方法がないことが原因である。HDD43を管理しているOS(特にこの場合ファイルシステム)を利用する際に、HDD43とI/Oするデータを格納するための論理アドレスが必要となる。OSは、システム用の仮想アドレス空間とアプリケーション毎の仮想アドレス空間とを用いてメモリ管理しているため、論理アドレスはアプリケーション毎に独立して存在している。CODECボ

ード41上のメモリ41aは物理アドレスをもつが、通常、ドライバによってシステム用（カーネル）の仮想アドレス空間にマップされ、カーネルモードの論理アドレスで利用される。また、アプリケーションはOSによって独自の論理アドレス空間が用意され、カーネルモードの論理アドレスへ直接アクセスすることはできない。

【0032】そこで、本発明ではシステムを構成するプログラムとして、以下の3つの処理を用意することで、CODECボード41からSCSI HDAカード42へのデータ転送を実現する。

(1) ボードに搭載されたメモリの、物理アドレスを論理アドレスへ変換する処理

(2) ある論理アドレス空間内の論理アドレスを別の論理アドレス空間に変換する処理

(3) ファイルへのリードやライトを行う処理

これらの処理機能は、所定のソフトウェアをCPU11に実行させることで実現する。

【0033】図3は、データ転送を実現するのに必要な処理機能を示す図である。本発明の実現に必要な処理機能は、OS60、デバイスドライバ70及びアプリケーション80によって提供される。

【0034】OS60は、仮想アドレス空間を、システム制御用のアドレス空間（カーネルモード仮想アドレス空間）とユーザアプリケーションが実行されているプロセス固有のアドレス空間（ユーザモード仮想アドレス空間）に分けて管理している。また、このOS60は、複数のプロセス（タスク）を同時に進行させるためのマルチタスク機能を有している。同時に進行する各プロセスにはユーザモード仮想アドレス空間が与えられ、各プロセスはユーザモード仮想アドレスを自由に使用することが30 できる。

【0035】また、OS60には、メモリマップ管理部61とファイル管理システム62が設けられている。メモリマップ管理部61は、実メモリの物理アドレスの仮想アドレス空間への写像を管理している。このような写像は、デバイスドライバ70からの要求に応じて実行することができる。ファイル管理システム62は、HDD43等へのファイルの入出力を管理する。例えば、ディレクトリ（フォルダ）構造を管理したり、格納すべきファイルへのファイル名や更新時刻（タイムスタンプ）の付加といった操作を行う。WindowsNTにおいてはNTFS (NT File System)がファイル管理システムに該当する。

【0036】デバイスドライバ70は、第1のアドレス変換要求部71と第2のアドレス変換要求部72とを有している。第1のアドレス変換要求部71は、CODECボード41に搭載されたメモリ41aの物理アドレスを論理アドレスへ変換するための処理要求を出力する。これは、例えばデバイスドライバ70がOS60に組み込まれる際に実行する。通常、CODECボード41上40

に存在するメモリ41aをOS60に登録する処理がこれにあたる。この段階では、あくまでもOS60のカーネルモード仮想アドレス空間にマップされており、この状態では、ユーザモードで実行されるアプリケーションから直接CODECボード41上に存在するメモリ41aの論理アドレスを指定することはできない。

【0037】第2のアドレス変換要求部72は、カーネルモード仮想アドレス空間内の論理アドレスを、ユーザモード仮想アドレス空間に変換するための要求をOS60に対して行う。具体的には、カーネルモード仮想アドレス空間に写像されたメモリ41aのメモリ空間を、アプリケーション80から利用可能なユーザモード仮想アドレス空間に変換する。処理の結果、アプリケーション80はCODECボード41上のメモリ41aのアドレスを知ることになる。

【0038】アプリケーション80は、映像データのノンリニア編集を行う。その機能の一部として、データの転送要求を行う転送要求部81を有している。転送要求部81は、例えば、スレッドにて実現している。ファイルのリードやライトのタイミングは、システム全体の動作の流れに則して行われる。例えば、アップロード時、CODECボード41上のメモリ41aいっぱい映像データが蓄積された時点でHDD43に書き込むなら、デバイスドライバ70に対してハードウェア割り込みによって要求を通知する。詳細をメッセージの形で通知することで、アプリケーション80は、どのファイルに、どのアドレスからどれくらい読み出すことが必要か知る。ダウンロード時は、メッセージのデータから、どのファイルのどの位置のデータを、どれくらいどのアドレスへ書き込むことが必要か知る。メッセージに基づき、アプリケーション80は「ReadFile writeFile」といったファイルI/O関数をコールする。バッファアドレスとして、デバイスドライバ70の第2のアドレス変換要求部72が獲得したアドレスとメッセージ内容から計算したアドレスを利用する。

【0039】これらの処理の流れを以下に示す。図4は、データ転送処理手順を示す図である。

【S1】デバイスドライバ70の第1のアドレス変換要求部71からの要求を受けたメモリマップ管理部61が、ボード上のメモリの物理アドレスを、OS60のカーネルモード仮想アドレスに変換する。この処理の関数コールの流れは、図5に示す。

【S2】アプリケーション80の転送要求部81からの要求を受けたファイル管理システム62が、デバイスオブジェクトをオープンする。OS60がWindowsNTであれば、転送要求部81が「CreateFile」関数をコールすることで実現できる。

【S3】アプリケーション80の転送要求部81からの要求を受けたファイル管理システム62が、I/Oを行うファイルオブジェクトをオープンする。これは、OS

60がWindowsNTであれば、転送要求部81が「CreateFile」関数をコールすることで実現できる。

〔S4〕デバイスドライバ70の第2のアドレス変換要求部72からの要求を受けたメモリマップ管理部61が、カーネルモード仮想アドレス空間の論理アドレスを、アプリケーション80が参照可能なユーザモード仮想アドレス空間内の論理アドレスに変換する。この処理の関数コールの流れは、図6に示す。

〔S5〕アプリケーション80の転送要求部81からの要求を受けたファイル管理システム62が、実際のデータ転送をSCSI HBAカード42に指令する。この処理の関数コールの流れは、図7に示す。

〔S6〕アプリケーション80の転送要求部81からの要求を受けたファイル管理システム62が、アドレスを解放する。これは、OS60がWindowsNTであれば、デバイスドライバ70が指定したフラグを用いた「DeviceControl」関数を、転送要求部81がコールすることで実現できる。

〔S7〕デバイスドライバ70の第2のアドレス変換要求部72からの要求を受けたメモリマップ管理部61が、獲得したボード上メモリ用論理アドレスを解放する。これは、OS60がWindowsNTであれば、アドレス変換要求部72が「ZnUnmapViewOfSection」関数をコールすることで実現できる。

〔S8〕アプリケーション80の転送要求部81からの要求を受けたファイル管理システム62が、デバイスオブジェクトをクローズする。これは、OS60がWindowsNTであれば、転送要求部81が「CloseHandle」関数をコールすることで実現できる。

〔S9〕アプリケーション80の転送要求部81からの要求を受けたファイル管理システム62が、ファイルオブジェクトをクローズする。これは、OS60がWindowsNTであれば、転送要求部81が「CloseHandle」関数をコールすることで実現できる。

〔0040〕図5は、カーネルモード仮想アドレス空間への交換手順を示す図である。なお、この処理は全て、デバイスドライバ70の第1のアドレス変換要求部71からの要求を受けたメモリマップ管理部61が実行する処理である。

〔S11〕ボードを検出する。これは、OS60がWindowsNTであれば、第1のアドレス変換要求部71が「HalGetBusData」関数をコールすることで実現できる。

〔S12〕物理アドレスを論理アドレスに変換する。これは、OS60がWindowsNTであれば、第1のアドレス変換要求部71が「HalTranslateBusAddress」関数をコールすることで実現できる。

〔S13〕OS60に論理アドレスとサイズを登録する。これは、OS60がWindowsNTであれば、第1のアドレス変換要求部71が「IoReportResourceUsage」関数をコールすることで実現できる。

〔0041〕図6は、ユーザモード仮想アドレス空間への交換手順を示す図である。なお、この処理は全て、デバイスドライバ70の第2のアドレス変換要求部72からの要求を受けたメモリマップ管理部61が実行する処理である。

〔S21〕セクションオブジェクトを新規作成する。これは、OS60がWindowsNTであれば、第2のアドレス変換要求部72が「ZwOpenSection」関数をコールすることで実現できる。

〔S22〕セクションをメモリマップに組み込む。これは、OS60がWindowsNTであれば、第2のアドレス変換要求部72が「ZwMapViewOfSection」関数をコールすることで実現できる。

〔S23〕セクションオブジェクトを解放する。これは、OS60がWindowsNTであれば、第2のアドレス変換要求部72が「ZwClose」関数をコールすることで実現できる。

〔0042〕図7は、データ転送処理手順を示す図である。なお、この処理は全て、アプリケーション80の転送要求部81からの要求を受けたファイル管理システム62が実行する処理である。

〔S31〕変換されたアドレスを問い合わせる。これは、OS60がWindowsNTであれば、転送要求部81が「DeviceIoControl」関数をコールすることで実現できる。

〔S32〕獲得したアドレスをバッファアドレスとして、ファイルのI/Oを行う。これは、OS60がWindowsNTであれば、転送要求部81が「ReadFile」もしくは「WriteFile」関数をコールすることで実現できる。

〔0043〕以上のような処理が行われることで、CODECボード41内のメモリ41aとHDD43との間でDMA転送を行うことができる。ここで、アドレス変換の様子を概念的に説明する。

〔0044〕図8は、アドレス空間のイメージを示す図である。このノンリニア編集機に実装されているOSは、以下の3つのアドレス空間を持っている。

- ・ Physical Memoryを表す物理アドレス空間
- ・ OSのカーネル（OSの基本部分）が利用するシステム空間（カーネルモード仮想アドレス空間）
- ・ アプリケーションそれぞれがもつ仮想アドレス空間（ユーザモード仮想アドレス空間）

これら3つのアドレス空間のうち、実体としてのメモリは物理アドレス空間にしかない。ほかの2つの空間は仮想メモリ空間で、実体としての物理アドレス空間に写像される。

〔0045〕あるアプリケーションが獲得したバッファ（ユーザモード仮想アドレス空間にある）は、実体は物理アドレス空間にセグメント（1単位で転送される固定長のブロック）単位で散在する。これは、物理メモリを各仮想メモリにマップする仮想メモリシステムが、ある

セグメント単位で管理して割り当てているためである。この1つのセグメントをページという。連続した仮想アドレスをもつバッファ（例えば、CODECボード41のメモリ41a）も実は、ページ単位で分散して物理アドレス上に存在する。

【0046】今回の発明では、CODECボード41のメモリ41aの物理アドレスをユーザモード仮想アドレス空間上に定義するために、WindowsNTに用意されているSectionオブジェクトを使用した。このSectionオブジェクトは、カーネルモード仮想アドレス空間のメモリを、あるアプリケーションのユーザモード仮想アドレス空間に割り当てるための管理情報となる構造体である。ユーザモード仮想アドレス空間には複数のアプリケーションが動いているが、Sectionオブジェクトは、この中の1つのアプリケーションのためのユーザモード仮想アドレス空間に割り当てることができる。なお、カーネルモード仮想アドレス空間を使った場合、その空間をユーザモード仮想アドレス空間にマップしても、物理アドレスは、物理アドレス空間内にページ単位で散在することとなる。これは、メモリに格納しきれないデータをHDDに格納しておき、必要なデータだけをページ単位でメモリに格納する処理（スワップ）を行っているからである。

【0047】ところが、CODECボード41などのボード上のメモリ41aでは、必要なデータが全てメモリ41a内に格納されているため、スワップを行う必要がない。従って、ボード上のメモリには連続した物理アドレスが割り当てられ、カーネルモード仮想アドレス空間内に連続したアドレスで割り当てることができる。このため、Sectionオブジェクトを使うことで、カーネルモード仮想アドレス空間内の連続したアドレスとして、カードの上のメモリの物理アドレスをマッピングできるようになる。アプリケーションのユーザモード仮想アドレス空間に連続したアドレスとしてマッピングされれば、アプリケーションからは、そのアドレスの先頭とデータ量などを指定し、1回の指令で全てのデータの転送要求を出すことが可能となる。連続したアドレスの大量のデータ転送であれば、SCSI HBAカード42のDMA転送機能を用いて、ボード間のDMA転送が行われる。

【0048】以上のように、ボード上のメモリのアドレスを、カーネルモード仮想アドレス空間からユーザモード仮想アドレス空間へ変換したことにより、アプリケーション80からボード上のメモリのデータを直接指定したデータ転送要求を出すことができる。すると、ファイル管理システム62の管理機能の提供を受けつつ、ボード間のDMA転送を実行することが可能となる。その結果、データ転送を効率よく行うことができる。

【0049】なお、データ転送のためのファイルのオープンは、例えばリアルタイム処理などを行う前の段階で実行する。WindowsNT4.0の場合、オープン時の指定とし

て、「NO#BUFFER」を必ず行う必要がある。これは、デフォルト（NO#BUFFER未指定）の場合、ファイルI/Oの際、OSが内部で用意したメモリを介してHDDとデータのやり取りを行ってしまうためである。

【0050】次に、本発明の応用例について説明する。図9は、ノンリニア編集機の別のハードウェア構成例を示す図である。これは、バスブリッジによって切り離されたバスに個別のSCSI HBAカードを設けることで、データ転送効率を向上させたものである。

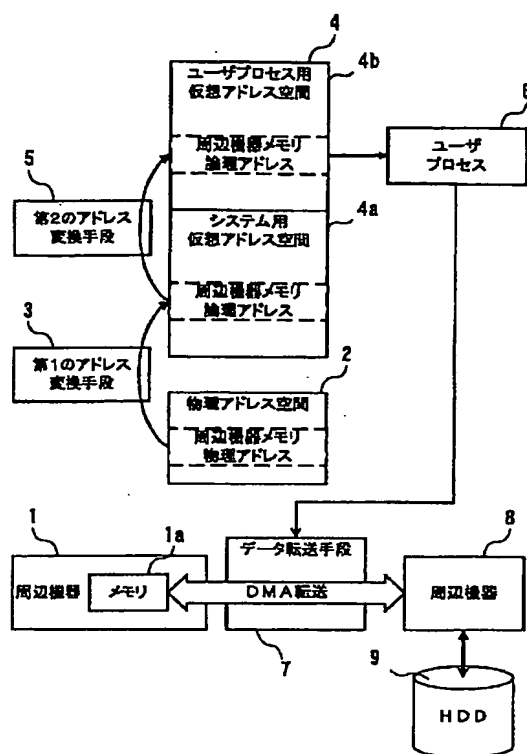
【0051】この例では、CPU101はPCIバスコントローラ102を介してシステムメモリ103とPCIバス121に接続されている。PCIバス121は、バスブリッジ104を介して別のPCIバス122に接続されている。さらにPCIバス122は、別のバスブリッジ105、106を介してPCIバス123、124に接続されている。PCIバス123には、CODECボード107とSCSI HBAカード108が接続されており、PCIバス124には、オーディオボード110とSCSI HBAカード111とが接続されている。SCSI HBAカード108には、映像データが格納されたHDD109が接続されており、SCSI HBAカード111には、音声データが格納されたHDD112が接続されている。

【0052】このようにシステム内におけるボードのレイアウトを工夫することで、システム全体のバス帯域を効率的に利用できる。具体的には、並行して行われるデータ転送間で相互に影響を与えないレイアウトを実現できる。図9におけるレイアウト図中、破線で囲まれたブロックが、1つのデータ伝送単位を形成する。

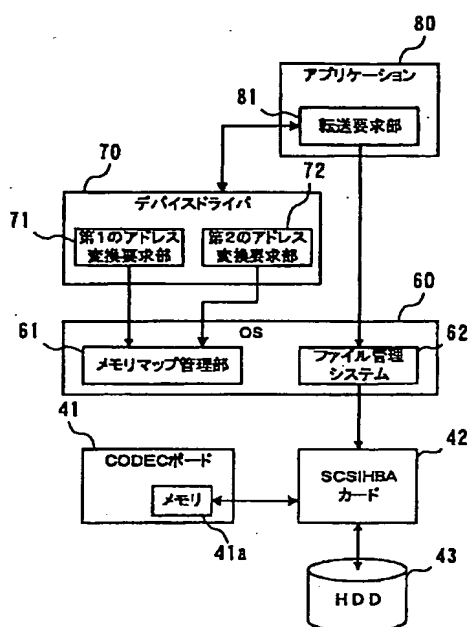
【0053】ある1つのバスブリッジ（またはPCIバスコントローラ）の二次バス側に2組以上のデータ転送が行われる状況では、バスのアービトレーション等余計な時間を消費し、結果としてデータ転送レートに悪影響を与える。そこで、この例では、データ転送単位として、CODECボード107とSCSI HBAカード108、オーディオボード110とSCSI HBAカード111の2単位を考える。ノンリニア編集におけるアップロード時やダウンロード時など、オーディオボード110もCODECボード107もそれぞれ、並行してHDD109、112からデータをリードしたりライトしたりする。

【0054】音声と映像では、データ要求のタイミングがまったく同期できず、それぞれの処理において必要なタイミングで要求され、処理する必要がある。そこで、CODECボード107と映像データが入っているHDD109が接続されているSCSI HBAカード108を1単位、オーディオボード110と音声データが入っているHDD112が接続されているSCSI HBAカード111を1単位としてレイアウトを工夫し、本発明を利用した伝送方法を用いることで、それぞれの単位に

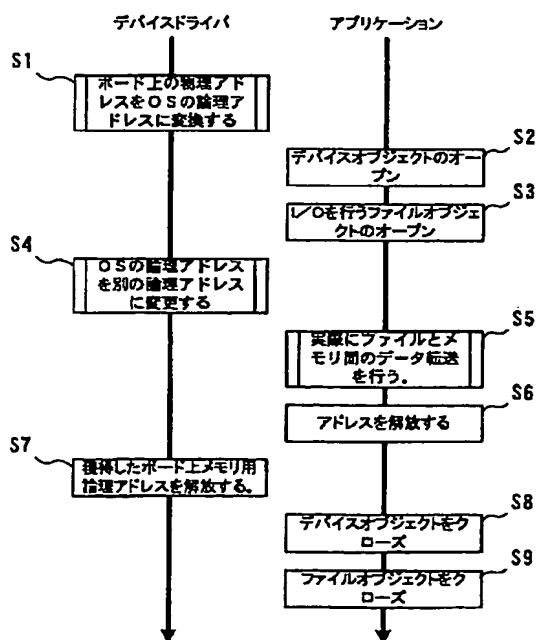
【図1】



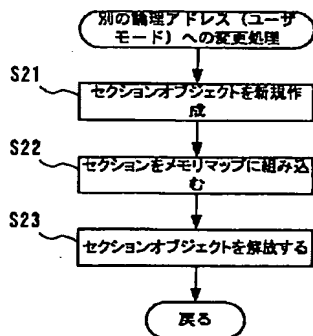
【図3】



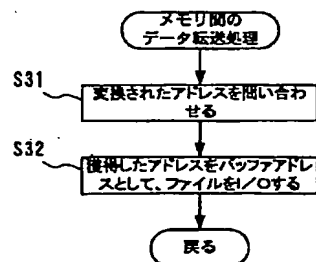
【図4】



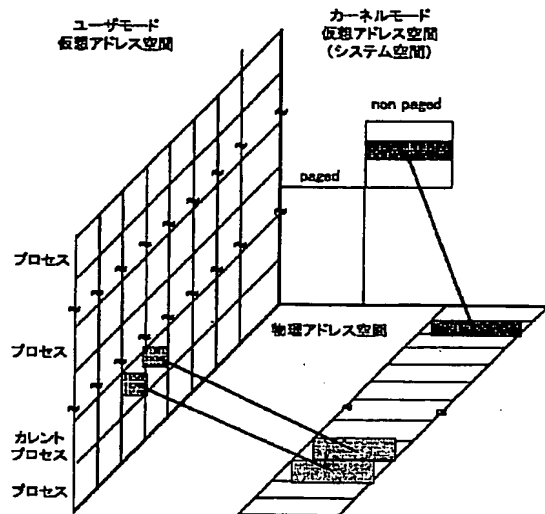
【図6】



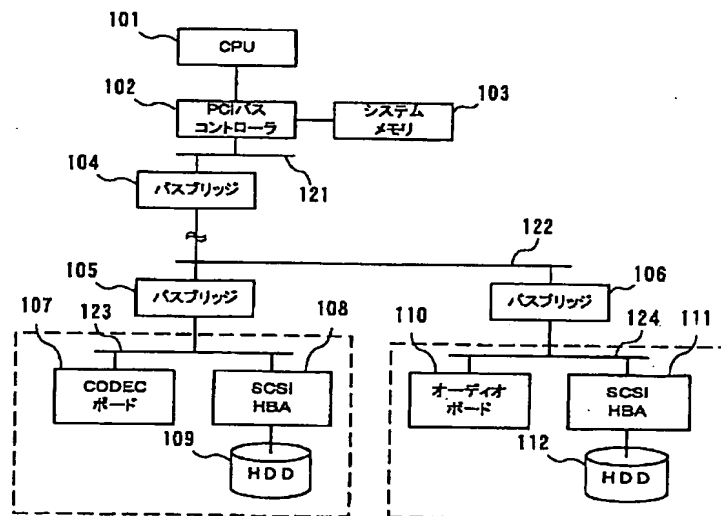
【図7】



【図8】



【図9】



フロントページの続き

(51)Int.Cl.
G11B 20/10

識別記号

F I
G11B 20/10

テーマコード (参考)
D

F ターム(参考) SB005 JJ12 KK16 LL15 MM31 RR01
TT11
SB014 EA03 HB27
SB060 AB26 AC11
SB061 BA03 DD01 DD11 GG02
SD044 AB01 AB05 BC01 CC04 HL02
HL11

* NOTICES *

JPO and NCIPi are not responsible for any damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.
2. **** shows the word which can not be translated.
3. In the drawings, any words are not translated.

DETAILED DESCRIPTION

[Detailed Description of the Invention]

[0001]

[Field of the Invention] This invention relates to the data processor for which a lot of data transfer is especially needed about the data processor which performs data transfer between two or more boards.

[0002]

[Description of the Prior Art] It is possible to replace conventionally the activity which was being done with the equipment of dedication by general-purpose computer by improvement in the engine performance of a computer. A video editor is in one of such the devices. That is, it is possible to perform the editing task of images, such as VTR (Video Tape Recorder), or voice using a computer. In that case, image information and speech information are stored in the hard disk drive unit of a computer, and an editing task is performed using the application program of a computer. Such a video edit system is called non-linear-editing machine.

[0003] The non-linear-editing machine is superior to the conventional edit system in that deformation and composition of an image can be performed freely, and the point that there is no degradation of image quality even if it repeats edit. For example, it is also easy for the image of an on-the-spot photo to compound CG (Computer Graphics). By the way, in a non-linear-editing machine, since the video data etc. is stored in a hard disk drive unit, it will be necessary through a general-purpose bus (for example, PCI (Peripheral Component Interconnect) bus) to transmit to other boards (for example, CODEC (coder-decoder) board) from a hard disk. Thus, when performing data transfer between boards, fundamentally, the system memory in the personal computer part in a nonlinear editor (non real-time control section) is made to intervene, and data transfer is performed. That is, two transfer processings, the transfer to a system memory from the board used as a data source and the transfer to the board which serves as a data target from a system memory, are performed. This approach is considered as the 1st conventional approach.

[0004] Moreover, there is also a method of transmitting data between direct boards without being influenced of software, such as OS (Operating System). A DMA (Direct Memory Access) transfer is used for this. Although a DMA transfer is the function in which it is used in case data transfer is fundamentally performed to a secondary storage from main storage (system memory), it is applying this to the data transfer between boards, and becomes accelerable [data transfer]. This approach is considered as the 2nd conventional approach.

[0005]

[Problem(s) to be Solved by the Invention] However, by the conventional approach, there were the following troubles also in which approach.

[0006] By the 1st conventional approach, since it is necessary to perform data transfer twice while the services (especially file management, memory management, etc.) which software (especially OS) offers can be received, the engine performance will become low.

[0007] By the 2nd conventional approach, although the transfer engine performance is high in order to transmit directly, the service (especially file management) which software (especially OS) offers cannot

be received. Therefore, in data transfer without need, such as file management, it is effective (for example, when transmitting image data to a graphic control board), but when saving to a hard disk, it cannot apply.

[0008] Therefore, the 1st conventional approach will be used in the system by which service of software (especially OS) becomes indispensable. However, since the very high transfer engine performance is required on the relation which transmits image data in a non-linear-editing machine, by the 1st conventional approach, it is unsuitable. That is, in the system which performs image processing like a non-linear-editing machine, service of software (especially OS) can be used and it is desired for the fast transfer between boards to be possible.

[0009] It aims at offering the data processor in which the high-speed data transfer between peripheral devices is possible, this invention being made in view of such a point, and receiving functional offer of file management system.

[0010]

[Means for Solving the Problem] In the data processor to which the peripheral device which carried memory was connected in order to solve the above-mentioned technical problem in this invention The 1st address translation means which changes the physical address of the memory of said peripheral device into the logical address in the virtual address space for system controls, The 2nd address translation means which changes the logical address after conversion by said 1st address translation means into the logical address in the virtual address space of the proper corresponding to the specific user process currently performed on the operating system, It is based on the data transfer demand which specified the logical address changed by said 2nd address translation means from said user process. The data processor characterized by having a data transfer means to perform data transfer by the Direct Memory Access between the memory of said peripheral device and other peripheral devices is offered.

[0011] According to such a data processor, the physical address of the memory of a peripheral device is changed into the logical address in the virtual address space for system controls by the 1st address translation means. Next, the logical address after conversion by the 1st address translation means is changed into the logical address in the virtual address space of the proper corresponding to the specific user process currently performed on the operating system by the 2nd address translation means. And if the data transfer demand which specified the logical address changed by the 2nd address translation means from an user process is advanced, data transfer by the Direct Memory Access between the memory of a peripheral device and other peripheral devices will be performed by the data transfer means.

[0012]

[Embodiment of the Invention] Hereafter, the gestalt of operation of this invention is explained with reference to a drawing. Drawing 1 is the principle block diagram of this invention. The peripheral device 1 which carried memory 1a is connected to the data processor of this invention. The address in a physical address space 2 is assigned to memory 1a. The 1st address translation means 3 is changed into the logical address of virtual-address-space 4a for systems in which the physical address of memory 1a of a peripheral device 1 was established in virtual address space 4. The 2nd address translation means 5 changes the logical address after conversion by the 1st address translation means 3 into the logical address in virtual-address-space 4b for user processes corresponding to the specific user process 6 currently performed on OS. The data transfer means 7 performs data transfer by the Direct Memory Access between memory 1a of a peripheral device 1, and other peripheral devices 8 based on the data transfer demand which specified the logical address changed by the 2nd address translation means 5 from an user process 6. In the example of drawing, the hard disk drive unit (HDD) 9 is connected to the peripheral device 8, and processing which writes the data of memory 1a in HDD9 is performed.

[0013] According to such a data processor, the address of memory 1a of a peripheral device 1 is defined in the virtual address space for user processes. Therefore, the data transfer demand which specified the address of memory 1a of a peripheral device 1 directly is attained from an user process 6. If it is the data transfer based on addressing from an user process 6, the function on OS by which the user process is performed is enjoyable. For example, when it stores data in HDD9 like the example of drawing, the

transmitted data are saved under management of the file management system of OS. Moreover, since the data transfer means 7 is performing the DMA transfer directly (main memory of a system is not minded) from memory 1a of a peripheral device 1, high data transmission efficiency can be obtained.

[0014] By applying this invention to a non-linear-editing machine, the fast transfer of image data becomes possible. Then, the example at the time of applying this invention to a non-linear-editing machine is shown below. In addition, in the following examples, while the example at the time of using WindowsNT (trademark of U.S. Microsoft Corp.) as an OS is shown, it explains.

[0015] Drawing 2 is drawing showing the example of a hardware configuration of a non-linear-editing system. CPU (Central Processing Unit) 11 performs the device driver mounted in OS, and application software performed on OS while performing the function of OS by controlling other peripheral devices and system memories 13. In addition, this CPU 11 is referring to an address translation table, and has the function to change the logical address of virtual memory space into the physical address of memory 41a in a system memory 13 or the CODEC board 41.

[0016] The PCI bus controller 12 is a controller for controlling PCI bus 21 from the CPU 11 side. A system memory 13 is the main storage of a system, and the program performed by CPU 11 and various data required for processing are stored temporarily.

[0017] PCI bus 21 is connected to two or more PCI buses 22-24 by the PCI bus bridges 31-33. Two or more PCI buses 21-24 are buses for performing data transfer between CPU 11 and a peripheral device, and data transfer between peripheral devices.

[0018] The PCI bus bridges 31-33 have the work which connects the PCI bus according to individual electrically mutually. Moreover, when data transfer between the equipment in one PCI bus is performed, control which does not output a signal is also performed to the PCI bus of another side.

[0019] The CODEC board 41 is connected to PCI bus 22, the local CPU bus 51, and the DAV (Digital Audio Visual) bus 52. This CODEC board 41 has the function which compresses and elongates image data. Moreover, memory 41a is prepared in the CODEC board 41, and the image data after processing of compression, elongation, etc. is stored in this memory 41a. the output image of the CODEC board 41 -- the DAV bus 52 -- minding -- DME (Distributed Management Environment) / SW (Switcher) board 45 -
- ** -- last **

[0020] The SCSIHBA (Small Computer System Interface Host Bus Adapter) card 42 is connected to PCI bus 22, the data sent through PCI bus 22 are stored in HDD43, or the data in HDD43 are transmitted to other devices through PCI bus 22. The SCSIHBA card 42 serves as a host of a PCI bus in the case of data transfer, and can control a DMA transfer.

[0021] HDD43 holds data, such as image data. It connects with PCI bus 23 and the local CPU bus 51, and local CPU board 44 carries local CPU. Local CPU controls actuation of the device (the example of drawing CODEC board 41) connected through the local CPU bus 51 on real time.

[0022] The DME/SW board 45 is connected to two or more I/O devices while connecting with PCI bus 24. The DME/SW board 45 performs processing to which effectiveness is applied or which changes an output destination change to image data. Moreover, the image input set up on this board is inputted into the CODEC board 41 through the DAV bus 52. Moreover, the image data sent from the CODEC board 41 relays the DME/SW board 45, and serves as a video output.

[0023] The local CPU bus 51 has connected between all the boards and local CPU boards 44 with which local CPU controls. And it is used for command transmission for local CPU to control other peripheral devices etc.

[0024] The DAV bus 52 is a bus for transmitting image data and voice data which were digital-signal-ized, and is connected to all the boards that process an image and voice data. In the example of drawing, the CODEC board 41, and the DME / SW board 45 are connected to the DAV bus 52.

[0025] In addition, in this example, although the non-linear-editing machine has the audio board etc. besides the element shown in drawing, since it extracted to the explanation about an image, it is not shown in drawing 2.

[0026] Next, the outline of upload processing is explained. Here, upload is recording an image and voice on HDD43 from input devices (a camera, videocassette recorder, etc.). It is as follows when the data

flow at the time of upload is explained briefly.

[0027] The image inputted through the DAV bus 52 is compressed on the CODEC board 41 (or incompressible), and is accumulated in memory 41a in the CODEC board 41. The data in memory 41a on the CODEC board 41 are recorded on HDD43 through the SCSIHBA card 42. HDD43 is formatted by OS and the data recorded are recorded on HDD43 as contents according to the file management system of OS.

[0028] Furthermore, the outline of download processing is explained. Here, download is reproducing to the storage device (a videocassette recorder and monitor) by which the image currently recorded on HDD43 and voice data were connected to the system. It is as follows when the data flow at the time of download is explained briefly.

[0029] The data currently recorded on HDD43 are managed and processed as a file from OS. When reproducing the recorded image data using the CODEC board 41, the contents of the file are read from HDD43. The read data are put on memory 41a on the CODEC board 41. And when the CODEC board 41 is compressed data, expanding processing is performed and it passes into the DAV bus 52. The data is received on the DME/SW board 45. The DME/SW board 45 pours the received image data to an output device.

[0030] Next, in case upload and download of data are performed, the processing for performing data transfer between the CODEC board 41 and the SCSIHBA card 42 is explained.

[0031] By the way, by the conventional approach, the data held at memory 41a of the CODEC board 41, for example on the occasion of upload were transmitted to the system memory 13, and the data was transmitted to the SCSIHDA card 42 after that. That is, the data of the CODEC board 41 were recorded on HDD43 by two transfers. Because, in the exchange with HDD43, it is because there is no method of specifying the physical address of direct CODEC board 41 top memory 41a. In case OS (file system in this case [Especially]) which has managed HDD43 is used, the logical address for storing the data which carry out I/O to HDD43 is needed. Since memory management of the OS is carried out using the virtual address space for systems, and the virtual address space for every application, the logical address exists independently for every application. Although memory 41a on the CODEC board 41 has a physical address, the map of it is carried out to the virtual address space for systems (kernel) by the driver, and it is usually used with the logical address of a kernel mode. Moreover, an original logical address space is prepared by OS and direct access of the application cannot be carried out to the logical address of a kernel mode by it.

[0032] Then, data transfer from the CODEC board 41 to the SCSIHDA card 42 is realized by preparing the following three processings as a program which constitutes a system from this invention.

(1) the processing which performs a lead and light of processing (2) processing (3) FAIRUHE which carries out logic ADORUSUHE conversion of the physical address of the memory carried in the board, and which changes the logical address in a certain logical address space into another logical address space -- the processing facility of these realizes predetermined software by performing CPU11.

[0033] Drawing 3 is drawing showing a processing facility required to realize data transfer. A processing facility required for implementation of this invention is offered by OS60, the device driver 70, and application 80.

[0034] OS60 divided virtual address space into the address space for system controls (kernel mode virtual address space), and the address space (user mode virtual address space) of a process proper where user application is performed, and is managed. Moreover, this OS60 has the multitasking feature for advancing multiple processes (task) to coincidence. User mode virtual address space is given to each process which advances to coincidence, and each process can use the user mode virtual address freely.

[0035] Moreover, the memory map Management Department 61 and file management system 62 are formed in OS60. The memory map Management Department 61 has managed the map to the virtual address space of the physical address of a real storage. Such a map can be performed according to the demand from a device driver 70. File management system 62 manages I/O of the file to HDD43 grade. For example, directory (folder) structure is managed or actuation of the file name to a file and the addition of modification time (time stump) which should be stored is performed. In WindowsNT, NTFS

(NT File System) corresponds to file management system.

[0036] The device driver 70 has the 1st address translation demand section 71 and the 2nd address translation demand section 72. The 1st address translation demand section 71 outputs the processing demand for carrying out logic ADORESUHE conversion of the physical address of memory 41a carried in the CODEC board 41. This is performed in case a device driver 70 is included in OS60. Usually, the processing which registers into OS60 memory 41a which exists on the CODEC board 41 hits this. In this phase, the map is carried out to the kernel mode virtual address space of OS60 to the last, and the logical address of memory 41a which exists on the direct CODEC board 41 from the application performed by the user mode cannot be specified in this condition.

[0037] The 2nd address translation demand section 72 gives the demand for changing the logical address in kernel mode virtual address space into user mode virtual address space to OS60. Specifically, the room of memory 41a mapped by kernel mode virtual address space is changed into available user mode virtual address space from application 80. Application 80 will know the address of memory 41a on the CODEC board 41 as a result of processing.

[0038] Application 80 performs non-linear editing of image data. As a part of the function, it has the transfer-request section 81 which performs a data transfer demand. The transfer-request section 81 is realized in the thread. Lead of a file and timing of a light are *(ed) with the flow of actuation of the whole system, and are performed. For example, if it writes in HDD43 at the time of upload when image data are stored to the limit of memory 41a on the CODEC board 41, hardware interrupt will notify a demand to a device driver 70. By notifying a detail in the form of a message, as for application 80, reading from which address to which file how much gets to know whether it is the need. the time of download -- the data of which location of the data of a message to which file -- how much -- which ADORESUHE writing **** -- things get to know whether it is the need. Based on a message, application 80 calls the file I/O function "ReadFile writeFile." As the buffer address, the address which the 2nd address translation demand section 72 of a device driver 70 gained, and the address calculated from the description are used.

[0039] The flow of these processings is shown below. Drawing 4 is drawing showing data transfer procedure.

The memory map Management Department 61 which received the demand from the 1st address translation demand section 71 of the [S1] device driver 70 changes the physical address of the memory on a board into the kernel mode virtual address of OS60. The flow of a function call of this processing is shown in drawing 5.

The file management system 62 which received the demand from the transfer-request section 81 of the [S2] application 80 opens a device object. If OS60 is WindowsNT, it is realizable because the transfer-request section 81 calls the "CreateFile" function.

The file management system 62 which received the demand from the transfer-request section 81 of the [S3] application 80 opens the file object which performs I/O. If OS60 is WindowsNT, this is realizable because the transfer-request section 81 calls the "CreateFile" function.

The memory map Management Department 61 which received the demand from the 2nd address translation demand section 72 of the [S4] device driver 70 changes the logical address of kernel mode virtual address space into the logical address in the user mode virtual address space which can refer to application 80. The flow of a function call of this processing is shown in drawing 6.

The file management system 62 which received the demand from the transfer-request section 81 of the [S5] application 80 orders the SCSIHBA card 42 actual data transfer. The flow of a function call of this processing is shown in drawing 7.

The file management system 62 which received the demand from the transfer-request section 81 of the [S6] application 80 releases the address. If OS60 is WindowsNT, this is realizable because the transfer-request section 81 calls the "DeviceControl" function using the flag specified by a device driver 70.

The memory map Management Department 61 which received the demand from the 2nd address translation demand section 72 of the [S7] device driver 70 releases the gained logical address for board top memory. If OS60 is WindowsNT, this is realizable because the address translation demand section

72 calls a "ZnUnmapViewOfSection" function.

The file management system 62 which received the demand from the transfer-request section 81 of the [S8] application 80 closes a device object. If OS60 is WindowsNT, this is realizable because the transfer-request section 81 calls the "CloseHandle" function.

The file management system 62 which received the demand from the transfer-request section 81 of the [S9] application 80 closes a file object. If OS60 is WindowsNT, this is realizable because the transfer-request section 81 calls the "CloseHandle" function.

[0040] Drawing 5 is drawing showing the conversion procedure to kernel mode virtual address space. In addition, the whole of this processing is the processing which the memory map Management Department 61 which received the demand from the 1st address translation demand section 71 of a device driver 70 performs.

[S11] board is detected. If OS60 is WindowsNT, this is realizable because the 1st address translation demand section 71 calls the "HalGetBusData" function.

[S12] physical address is changed into the logical address. If OS60 is WindowsNT, this is realizable because the 1st address translation demand section 71 calls a "HalTranslateBusAddress" function.

The logical address and size are registered into [S13] OS60. If OS60 is WindowsNT, this is realizable because the 1st address translation demand section 71 calls a "IoReportResourceUsage" function.

[0041] Drawing 6 is drawing showing the conversion procedure to user mode virtual address space. In addition, the whole of this processing is the processing which the memory map Management Department 61 which received the demand from the 2nd address translation demand section 72 of a device driver 70 performs.

[S21] section object is newly created. If OS60 is WindowsNT, this is realizable because the 2nd address translation demand section 72 calls the "ZwOpenSection" function.

[S22] section is included in a memory map. If OS60 is WindowsNT, this is realizable because the 2nd address translation demand section 72 calls the "ZwMapViewOfSection" function.

[S23] section object is released. If OS60 is WindowsNT, this is realizable because the 2nd address translation demand section 72 calls the "ZwClose" function.

[0042] Drawing 7 is drawing showing data transfer procedure. In addition, the whole of this processing is the processing which the file management system 62 which received the demand from the transfer-request section 81 of application 80 performs.

[S31] translated address is asked. If OS60 is WindowsNT, this is realizable because the transfer-request section 81 calls the "DeviceIoControl" function.

I/O of a file is performed by making into the buffer address the address which carried out [S32] acquisition. If OS60 is WindowsNT, this is realizable because the transfer-request section 81 calls "ReadFile" or the "WriteFile" function.

[0043] By the above processings being performed, a DMA transfer can be performed between memory 41a and HDD43 in the CODEC board 41. Here, the situation of address translation is explained notionally.

[0044] Drawing 8 is drawing showing the image of an address space. OS mounted in this non-linear-editing machine has the following three address spaces.

- System space which the kernel (a part for the radical headquarters of OS) showing Physical Memory of a physical address space and OS uses (kernel mode virtual address space)
- Virtual address space which each application has (user mode virtual address space)

There is memory as a stereo among these three address spaces only in a physical address space. Other two space is virtual memory space, and is mapped by the physical address space as a stereo.

[0045] For the buffer (it is in user mode virtual address space) which a certain application gained, stereos are scattered in a physical address space per segment (fixed-length block transmitted in one unit). This is because the virtual memory system which carries out a map is managing and assigning the physical memory to each virtual memory in a certain segment unit. This one segment is called page. In fact, it distributes per page and a buffer (for example, memory 41a of the CODEC board 41) with the continuous virtual address also exists on a physical address.

[0046] In this invention, in order to define the physical address of memory 41a of the CODEC board 41 on user mode virtual address space, the Section object currently prepared for WindowsNT was used. This Section object is the structure used as the management information for allocating the memory of kernel mode virtual address space to the user mode virtual address space of a certain application. A Section object can be assigned to the user mode virtual address space for one application in this although two or more applications are moving to user mode virtual address space. When kernel mode virtual address space is used, even if it carries out the map of the space to user mode virtual address space, physical addresses will be in addition, scattered per page in a physical address space. This is because processing (swap) which stores in HDD the data which cannot be stored in memory, and stores only required data in memory per page is performed.

[0047] However, in memory 41a on boards, such as the CODEC board 41, since all required data are stored in memory 41a, it is not necessary to swap. Therefore, the continuous physical address is assigned to the memory on a board, and it can assign in the address which continued in kernel mode virtual address space. For this reason, the physical address of the memory on a card can be mapped now by using a Section object as the address with which it continued in kernel mode virtual address space. If mapped as the address which followed the user mode virtual address space of application, a head, the amount of data, etc. of the address will be specified, and it will become possible from application to advance all data transfer demands by one command. If it is a lot of data transfer of the continuous address, the DMA transfer between boards will be performed using the DMA transfer function of the SCSIHBA card 42.

[0048] As mentioned above, the data transfer demand which specified the data of the memory on a board directly can be advanced from application 80 by having changed the address of the memory on a board into user mode virtual address space from kernel mode virtual address space. Then, it becomes possible to perform the DMA transfer between boards, receiving offer of the function manager of file management system 62. Consequently, data transfer can be performed efficiently.

[0049] In addition, opening of the file for data transfer is performed in the phase before performing a real-time operation etc. In WindowsNT4.0, it is necessary to surely perform "NO#BUFFER" as assignment at the time of opening. This is for performing an exchange of HDD and data through the memory which OS prepared inside in the case of file I/O in a default (NO#BUFFER un-specifying) case.

[0050] Next, the application of this invention is explained. Drawing 9 is drawing showing another example of a hardware configuration of a non-linear-editing machine. This is preparing the SCSIHBA card according to individual in the bus separated by the bus bridge, and raises data transmission efficiency.

[0051] In this example, CPU101 is connected with the system memory 103 through the PCI bus controller 102 at PCI bus 121. PCI bus 121 is connected to another PCI bus 122 through the bus bridge 104. Furthermore, PCI bus 122 is connected to PCI bus 123,124 through another bus bridge 105,106. The CODEC board 107 and the SCSIHBA card 108 are connected to PCI bus 123, and the audio board 110 and the SCSIHBA card 111 are connected to PCI bus 124. HDD109 in which image data were stored is connected to the SCSIHBA card 108, and HDD112 in which voice data was stored is connected to the SCSIHBA card 111.

[0052] Thus, with devising the layout of the board in a system, the bus band of the whole system can be used efficiently. The layout which does not specifically affect it mutually between the data transfer performed in parallel is realizable. The flocks surrounded with the broken line form one data transmission unit among the layout pattern in drawing 9.

[0053] In the situation that 2 or more sets of data transfer is performed to one certain secondary bus side of a bus bridge (or PCI bus controller), excessive time amount, such as an Arbitration of a bus, is consumed and it has a bad influence on a data transfer rate as a result. So, in this example, two units of the CODEC board 107, the SCSIHBA card 108 and an audio board 110, and the SCSIHBA card 111 are considered as a data transfer unit. In parallel, data are led from HDD109,112 or an audio board 110 and the CODEC board 107 carry out the light of the time of the upload in non-linear editing, and download

etc., respectively.

[0054] With voice and an image, timing of a data demand cannot synchronize at all, but is required to required timing in each processing, and needs to process. Then, a layout is devised by making into one unit the SCSIHBA card 111 by which HDD112 containing voice data is connected with one unit and an audio board 110 in the SCSIHBA card 108 by which HDD109 containing image data is connected with the CODEC board 107, and it can make the most of a bus in each unit by using the transmission approach using this invention. Consequently, there are the following advantages.

- the data transfer between a certain boards (the example of implementation between a CODEC board and a SCSIHBA card) -- setting -- the band of a bus -- the maximum -- it becomes available.
- Time amount which a transfer takes can be lessened (it realizes by 1/2 or less time amount).
- The computer resource (it is specifically the count capacity and the system memory of CPU) which is needed for data transfer decreases.
- In the system by which data transfer processing of N group is performed in parallel, the maximum data transfer capacity serves as band *N of a bus, and data transfer capacity improves.

[0055] In addition, the contents of processing of the function which the above-mentioned non-linear-editing machine should have are described by computer by the program recorded on the record medium which can be read, and the above-mentioned processing is realized by the computer by executing this program by computer (non-linear-editing machine). As a record medium which can be read, there are a magnetic recording medium, semiconductor memory, etc. by computer. When making it circulate to a commercial scene, store a program in portable mold record media, such as CD-ROM (Compact Disk Read Only Memory) and a floppy (trademark) disk, and they are circulated, or it stores in the storage of the computer connected through the network, and can also transmit to other computers through a network. In case it performs by computer, the program is stored in the hard disk drive unit in a computer etc., and it loads to main memory and performs.

[0056]

[Effect of the Invention] Since the address of the memory of a peripheral device was defined as having explained above in the virtual address space for user processes by this invention, the data transfer demand which specified the address of the memory of a peripheral device directly is attained from an user process. Consequently, the data transfer by the Direct Memory Access of peripheral devices becomes possible, enjoying the function on the operating system with which the user process is performed.

[Translation done.]